



# Optimizing Word2Vec Dimensions for Sentiment Analysis of Photomath Reviews using Random Forest and SVM

**Diva Azty Varissa Azis<sup>\*</sup>, Yuliant Sibaroni**

Informatic, School of Computing, Telkom University, Bandung, Indonesia

Email: <sup>1,\*</sup>divaazvarissa@student.telkomuniversity.ac.id, <sup>2</sup>yuliant@telkomuniversity.ac.id

Correspondence Author Email: <sup>1</sup>divaazvarissa@student.telkomuniversity.ac.id

Submitted: **04/01/2025**; Accepted: **26/02/2025**; Published: **01/03/2025**

**Abstract**—Technology in the Industrial Revolution 4.0 era supports modern learning through apps like Photomath, simplifying math problem-solving for users. However, diverse user reviews highlight the need for sentiment analysis to evaluate app quality. This research analyzes 9,059 reviews of Photomath collected from the Google Play Store using Python. Word2Vec is used in the study to compare Random Forest and Support Vector Machine (SVM) classifiers for feature extraction. To ensure clean and consistent data, preprocessing techniques such as stemming, tokenization, and stopword removal were used. Text with rich semantic aspects was mathematically represented using Word2Vec. The findings show that SVM using an RBF kernel performed better than Random Forest, with an F1-score of 88.5%, 88.5% accuracy, 88.7% precision, and 88.5% recall. Performance was effectively improved by combining 300-dimensional Word2Vec with stemming algorithms. While Random Forest achieved slightly lower accuracy, it shows promise for specific use cases. This study offers practical insights for improving Photomath by tailoring updates based on user sentiment. The findings emphasize the importance of preprocessing, dimensional optimization, and classifier selection in developing accurate sentiment analysis models. Limitations include the dataset size and the use of classical machine learning models. Future research could address these by exploring larger datasets or deep learning techniques to further improve performance.

**Keywords:** Photomath; Random Forest; Sentiment Analysis; Support Vector Machine (SVM); Word2Vec

## 1. INTRODUCTION

The industrialization of phase 4.0 has transformed modern life, with digital technologies becoming integral to various fields, including education [1]. The use of this technology not only supports the career field, but also serves as a means of learning. Overall, technology as a learning medium can be utilized through various applications that can be downloaded on mobile phones [2]. In the world of education, many utilize technology as a learning tool through applications that can help in learning and solving problems that are not easy to solve. One of them is math [3]. Among the various applications available, photomath is one of the most popular ones used to solve math problems. MicroBlink created Photomath in 2011. It uses advanced text reader technology to scan math problems that users upload [4]. The answer and solution steps will be displayed by this application. This application is very suitable for use as a reference for solving difficult problems [5]. However, in the field of education, this innovation has sparked both support and criticism. A number of user experiences can be seen and written in the photomath application review column [3]. User reviews can include advantages, disadvantages, or problems with the photomath application. This makes important information for user needs increasingly difficult to find because of the many opinions that arise. To address this issue, natural language processing (NLP) techniques enable computers to understand and process human language for tasks such as sentiment analysis. In this research, sentiment analysis is used as a computational study to find opinions, attitudes, and emotions towards an entity in a review column [6].

Sentiment analysis is a technique used to identify how a sentiment is expressed through text and how it can be categorized as positive or negative [7]. Sentiment analysis is usually used to evaluate opinions about an application with the aim of improving its quality in the future. In this context, sentiment analysis can be implemented on photomath application reviews [8]. In this research, there are several references that are relevant to previous research that focuses on sentiment analysis of application reviews. The methods used are SVM, Random Forest and Word2Vec. Some of these studies involve research [9] by M. J. Aufa and A. Qoiriah (2022) focused on sentiment analysis of users of the online learning platform Coursera using the Random Forest algorithm combined with Word2Vec feature extraction. After the filtering process, a total of 706,828 data entries consisting of 5 columns were obtained. The study achieved the highest accuracy with an 80:20 data proportion, a Word2Vec window size of 5, and a feature size of 100, resulting in a train score of 92.7% and a test score of 91.4%. Research [10] by F. A. Larasati et al. (2022) analyzed sentiment in application reviews using the Random Forest method. The study utilized data retrieval through web scraping techniques and Jupyter Notebook tools with the Google-Play-Scraper API. The data was split into 80% training and 20% testing datasets, achieving an accuracy of 84%, recall of 84%, F1-Score of 84%, and precision of 84% with a tree depth of 65 and 400 trees. This demonstrates the effectiveness of Random Forest in sentiment analysis tasks for application reviews. Research [11] by S. N. Adhan et al. (2024) analyzed the sentiment of Wattpad application reviews on the Google Play Store using the Random Forest method. The study was further optimized using the SMOTE (Synthetic Minority Oversampling Technique) to address class imbalance issues. The findings revealed that unbalanced sentiment classification tends to achieve better performance using the Random Forest method without SMOTE optimization. The reported metrics include an accuracy of 84.05%, precision of 84.71%, recall of 91.60%, F1-Score of 88.02%, FPR of 8.40%, and an AUC of 0.9166, indicating excellent classification performance. Random

forest was chosen because it is one of the effective classification algorithms for analyzing sentiment and there are advantages, namely the ability to improve accuracy results if there is missing data, be resistant to outliers, and be effective in data storage [12].

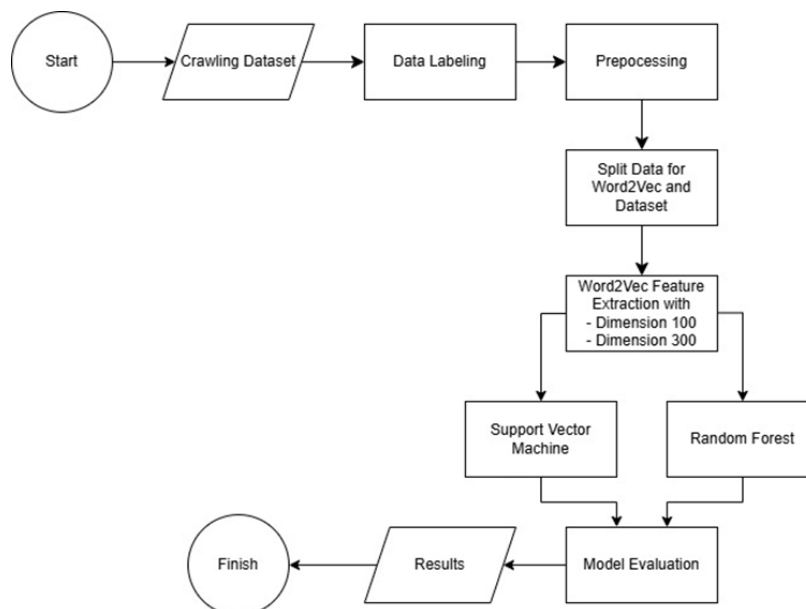
Research by M. A. A. Jihad et al. (2021) focused on sentiment analysis of movie reviews using Word2Vec and Support Vector Machine (SVM) [13]. The study applied lemmatization and utilized Word2Vec with 300 dimensions, combined with a linear SVM classification method, achieving the best accuracy of 78.75% and the highest F1-score of 78.74%. This research demonstrated that this approach provided superior performance compared to other tested methods. Research by D. I. Rifai (2024) focused on sentiment analysis of TikTok user reviews using Word2Vec models combined with the Support Vector Machine (SVM) method [14]. The study employed Word2Vec in the forms of CBOW and skip-gram to compare accuracy in sentiment classification. The CBOW model achieved an accuracy of 66%, while the skip-gram model performed slightly better with an accuracy of 68%. The data for this research was sourced from application user reviews on the Google Play Store. Research by P. E. Shopee and S. Watmah (2021) compared the performance of K-Nearest Neighbor (K-NN), Support Vector Machine (SVM), and Random Forest methods in classification tasks [15]. The study focused on evaluating the accuracy, precision, and recall of these classification techniques, finding that the SVM method outperformed the others with an accuracy of 89.4%, precision of 89.5%, and recall of 89.7%. Enquiry by Engineering (2023) analyzed sentiment towards the usage of the Shopee application using the Support Vector Machine (SVM) algorithm [16]. Collected 3000 review data points through a scraping procedure and achieved outstanding performance, with an accuracy rate of 98% and an F1-score of 98%. This demonstrates the potential of SVM for high-accuracy sentiment classification tasks. The ability of the SVM approach to reliably categorize evaluations into good, negative, and neutral groups led to its selection. Word2Vec was used because it could turn words into vectors. Word2Vec can assist in overcoming difficult sentiment analysis problems by comprehending the meaning of words in user reviews [17]. Although sentiment analysis has been extensively used in earlier research utilizing techniques like Word2Vec, Random Forest, and Support Vector Machine (SVM), the majority of these studies have not examined the effects of various Word2Vec dimensions on performance.

The results of earlier research indicate that the Random Forest and SVM approaches, when paired with Word2Vec feature extraction, exhibit highly effective sentiment analysis performance. Though this parameter can be changed to increase accuracy, prior research has tended to ignore the variance in Word2Vec dimension size. This study aims to analyze the sentiment of user reviews for the Photomath app using Random Forest and SVM methods, optimized with Word2Vec feature extraction. Different dimensions are tested to identify the most effective configuration. This research is expected to identify the best method in classifying review sentiment as positive or negative, as well as provide deeper insights into user experience to support the quality improvement of the Photomath app in the future.

## 2. RESEARCH METHODOLOGY

### 2.1 Research Phases

Examining the opinions of users in reviews of the Photomath app on the Google Playstore is the goal of this study. The following outlines the system that has been created in Figure 1.



**Figure 1.** Flowchart of system design



This figure illustrates the workflow for analyzing user opinions in reviews of the Photomath app from the Google Play Store. The process begins with crawling the dataset, which involves collecting user reviews from the platform. Following this, the reviews are subjected to data labeling, where the reviews are categorized into specific sentiment classes (positive and negative). The labeled data is then processed in the preprocessing stage to clean and prepare it for analysis. After preprocessing, the dataset is split into two components: one for Word2Vec feature extraction and the other for creating a dataset to train models. Two configurations are used for Word2Vec feature extraction: 100 and 300 dimensions. Two machine learning models, Random Forest (RF) and Support Vector Machine (SVM), use these vectorized features as input. Both models undergo training and performance evaluation. The Model Evaluation stage analyzes the evaluation findings to ascertain the models' efficacy. The process ends with a summary of the findings, which are subsequently utilized to learn more about user perceptions.

## 2.2 Crawling Dataset

Using Python for web scraping, the study's data came from user reviews of the Photomath app on the Google Play Store. 9,059 review entries from different time periods were collected, representing a diverse spectrum of user experiences. The dataset gives a fair picture of user sentiment because it contains both positive and negative evaluations. To guarantee data quality, specific filtering criteria were used before to collection. To concentrate only on evaluations with a definite positive or negative sentiment, neutral reviews were removed from the dataset. Different review lengths are displayed in the dataset; some evaluations are shorter and more succinct, while others include thorough commentary. Sentiment analysis must take into account the diversity of sentiment expression, which may be influenced by these differences in review duration. Table 1 gives a summary of the makeup of the dataset and shows the distribution of both favorable and negative ratings.

**Table 1.** Example of Dataset

Sentence
Sangat berguna untuk pelajar maupun orang tua yg lagi mengerjakan pr tapi ga tau caranya gimana. Sudah ada caranya diaplikasinya dan diberi penjelasan sangat mudah di mengerti. Pertahankan itu..
Jelek banget sih apk nya , masak udah discan beberapa kali katanya ngak pas , jawaban nya juga banyak yang bener !!!
Aplikasi ini kalo bisa dikasi bintang 1000 gw kasih Terima kasih juga buat developernya sgt membantu kami para murid yg agak kurang di bidang matematika :)

## 2.3 Data Labelling

One of the most important steps in getting ready for training a machine learning model is data labeling. For this study, every review in the dataset was given a sentiment-based score: reviews that were favorable received a rating of 5, and those that were negative received a value of 1. By classifying user thoughts into two main groups using this binary labeling technique, the model is able to distinguish between positive and negative feelings. Examples of how the reviews were categorized are displayed in Tables 2 and Tables 3. Neutral evaluations were not included in the dataset, therefore it is crucial to remember that the categorization procedure only looked at reviews that were obviously favorable or negative. This guarantees a more targeted user satisfaction study, which is essential for enhancing the Photomath app.

**Table 2.** Data Labelling

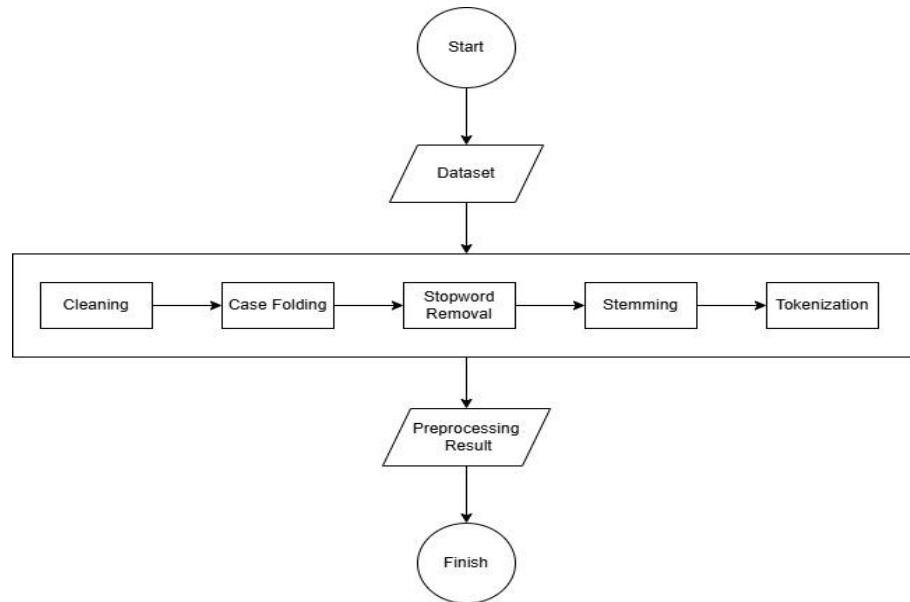
Label	Rating	Sentence
Positive	5	Sangat berguna untuk pelajar maupun orang tua yg lagi mengerjakan pr tapi ga tau caranya gimana. Sudah ada caranya diaplikasinya dan diberi penjelasan sangat mudah di mengerti. Pertahankan itu..
Negative	1	Jelek banget sih apk nya , masak udah discan beberapa kali katanya ngak pas , jawaban nya juga banyak yang bener !!!

**Table 3.** Data Labeling Results

Label	Rating	Review Count
Positive	5	5975
Negative	1	3084

## 2.4 Preprocessing

Preprocessing aims to address the issues identified during data processing. The steps involved in preprocessing include cleaning, case folding, stopword removal, stemming, and tokenization. Figure 2 illustrates the flowchart for the preprocessing stage.



**Figure 2.** Flowchart Preprocessing

This flowchart illustrates the preprocessing steps for the dataset. The process starts with:

a. Cleaning

Cleaning is a stage that must be done in preprocessing by removing punctuation, numbers, symbols, and double spaces that are not relevant to the research. A sample of the cleaning process is in Table 4.

**Table 4.** Sample of Cleaning

[! “ # \$ % & ' ( ) * + , - . / : ; < = > ? @ [ \ ] ^ _ ` {   } ~] and others
---

This table shows examples of elements removed during the cleaning stage of preprocessing. The cleaning process aims to improve data quality by removing irrelevant elements, such as punctuation marks (!, ?, ,), numbers (1, 2), special symbols (@, #, \$), and double spaces. By removing these elements, the text becomes cleaner and easier to process in subsequent stages of analysis, such as tokenization and feature extraction.

b. Case Folding

Case folding is a stage where all uppercase letters in a word are converted into lowercase letters. This helps eliminate differences in capital letters and makes the data more uniform for analysis.

c. Stopword Removal

Stopword removal is a process in which irrelevant or insignificant words are eliminated, reducing their impact on the classification process. In this research, the NLTK (Natural Language Toolkit), dictionary is used because NLTK can provide various languages. A sample of the stopwords removal process is in Table 5.

**Table 5.** Sample of Stopword Removal

yang, dan, di, dari, pada, dan, atau, tapi, karena, ini, itu, tersebut
--

This table shows examples of words removed during the stopwords removal stage of preprocessing. The stopwords removal process aims to improve classification efficiency and accuracy by removing words that appear frequently but have little meaning or do not provide significant information, such as conjunctions, pronouns, and prepositions. By removing these words, the model can focus more on words that are more informative and relevant for sentiment analysis or other text classification.

d. Stemming

Stemming is a process focused on transforming text into its base form or root word. This is accomplished through removing the word's affixes. Which are found in prefixes, suffixes, and combinations. The stemming process in Python for Indonesian can be done using the Sastrawi library, which is specifically designed for these needs. Sastrawi was chosen because it is specifically designed for Indonesian, providing more accurate and relevant results compared to common stemming tools such as NLTK or Porter. The selection of Sastrawi will increase the reproducibility of the study as this library is widely used in various Indonesian NLP research and is easy to implement with clear documentation. A sample of the stemming process is in Table 6.

**Table 6.** Sample of Stemming

Compound words	Base word
berguna	guna
mengerjakan	kerja

caranya	cara
penjelasan	jelas
pertahankan	tahan
nyambung	sambung
ditanyakan	tanya
membantu	bantu
diberi	beri
pembelajaran	ajar

This table shows examples of compound words that have gone through the stemming process to be converted into their base word form. The stemming process aims to simplify the text by reducing words to their base form, thus facilitating data analysis and improving the accuracy of the classification model. By performing stemming, the morphological variation of words can be reduced, allowing the model to process different words with the same meaning as a single entity, ultimately improving the efficiency and accuracy of the text classification process.

e. Tokenization

Tokenization is a stage where the sentence will be broken down into words (tokens). The goal is to simplify the classification process. A sample of the tokenization process is in Table 7.

Table 7. Sample of Tokenization

Before	After
sangat guna pelajar orang tua kerja pr cara sudah cara aplikasi beri jelas sangat mudah ngerti tahan	‘sangat’, ‘guna’, ‘pelajar’, ‘orang’, ‘tua’, ‘kerja’, ‘pr’, ‘cara’, ‘sudah’, ‘cara’, ‘aplikasi’, ‘beri’, ‘jelas’, ‘sangat’, ‘mudah’, ‘ngerti’, ‘tahan’

This table shows an example of the tokenization process where a sentence is broken down into individual words (tokens). The tokenization process aims to simplify the text into smaller units that can be analyzed individually by the classification model. By tokenizing, the text becomes more structured, allowing the model to handle each word separately and improving accuracy in understanding and processing the text during sentiment analysis or other classification.

## 2.5 Split Data

After the preprocessing stage is complete, data separation is performed as the next step. At this phase of development, the dataset will be assigned to sections for testing and training. In this process, the data is divided into two methods based on the needs. To train the word model, 8059 of the 9059-review data are designated as Word2Vec Corpus. Later, the prediction model will be trained and tested using the remaining 1000 data.

## 2.6 Word2Vec Feature Extraction

Mikolov first proposed the Word2Vec method in 2013, which takes a corpus as input and produces a vector as output. Each unique word can be converted into a vector by Word2Vec. One of the advantages of Word2Vec is the ability of the obtained vector to represent the context similarity between two words. Word2Vec is regarded as an efficient algorithm for representing word vectors in natural language processing (NLP), as it clusters similar words into the same vector. It obtains the representation of words into vectors using a neural network, producing vectors in a dimensional space that characterize the words' semantic meaning [18].

Continuous Bag of Words (CBOW) and Skip-gram are the two primary architectures that make up the Word2Vec paradigm. CBOW forecasts the target word by considering its context. This method requires less training time and tends to provide slightly better accuracy for more frequent words. On the other hand, the Skip-gram approach predicts the context around a single word. Skip-gram makes good use of the little training data, and it can better represent words that don't appear often [18]. CBOW and skip-gram architecture models can be seen in Figure 3.

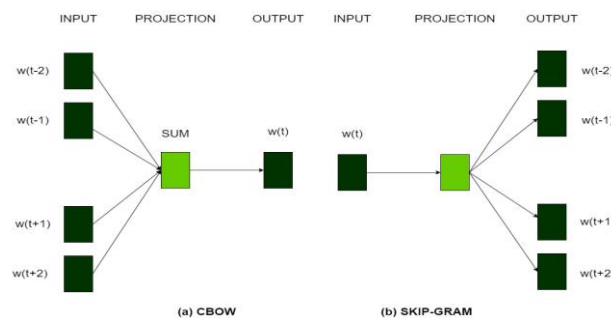


Figure 3. CBOW and Skip-Gram Model Architecture

Based on the trade-off between computing speed and word representation quality, this study compares Word2Vec's 100 and 300 dimensions for sentiment analysis of app reviews. While the 300 dimension offers a richer and more detailed representation of words, which is advantageous for capturing the subtleties of language in larger and complex datasets, the 100 dimension offers a lighter and faster representation, which is appropriate for small datasets and helps lower the risk of overfitting [18]. In this dissertation, features are extracted from preprocessed and data-split data using the Skip-Gram version of the Word2Vec algorithm. Because it can capture rich semantic nuances and broader context linkages, skip-gram is used. Example Figure 4 shows the Word2Vec model with Skip-gram.

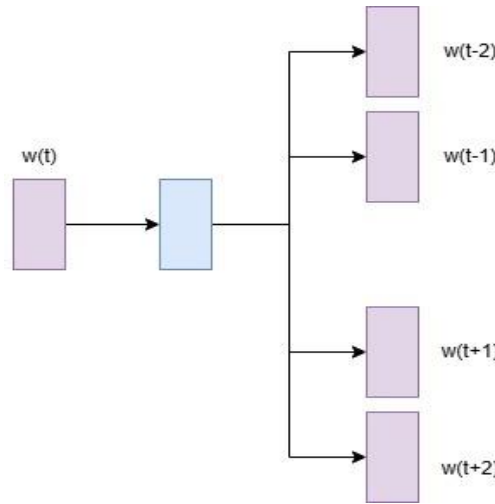


Figure 4. Skip-Gram Model Architecture

## 2.7 Support Vector Machine

At this point, SVM is utilized to identify a hyperplane to differentiate between positive and negative classes following data processing and Word2Vec feature extraction. SVM is a classification method that applies supervised learning to predict classes based on existing patterns obtained during the training process. The line that is furthest distant from the nearest training data point across all classes is, in fact, the ideal dividing line. Generally, a larger margin reduces the likelihood of generalization errors in the separator. The distance between a class's vector points and the hyperplane is known as the margin [17]. Referring to this, Figure 5 is an image of the hyperlane representation in the SVM model.

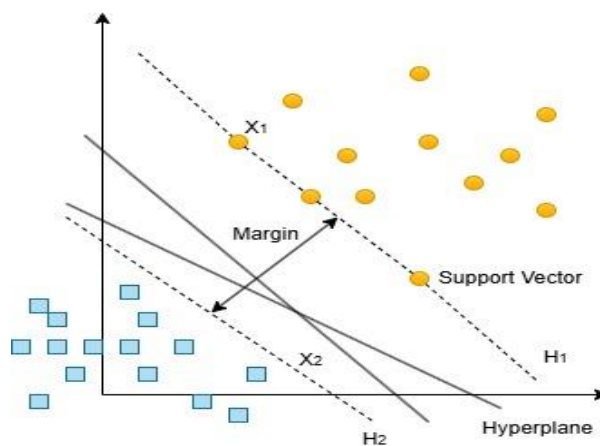


Figure 5. Hyperlane representation of SVM Model

In SVM, there are several types of kernels that can be used [19]:

### a. Linear Kernel

In the analysis using a linear kernel, the adjusted parameter is C or Cost. Optimization of parameter C is generally done through trial-and-error method. The following is the linear kernel equation.

$$K(x_i, x) = X_i^T X \tag{1}$$

The linear kernel function  $K(x_i, x) = X_i^T X$  It is utilized to assess the similarity between two input vectors,  $x_i$  and  $x$ . In this calculation,  $X_i^T$  represents the transpose of the vector  $x_i$ , which converts a row vector into a column vector (or vice versa). The result of the operation  $X_i^T X$  is the dot product between the two vectors, producing a

scalar value. This value indicates the linear similarity between the vectors, where a higher value suggests that the vectors are more similar or oriented in the same direction in the feature space.

b. RBF Kernel

When the analysis is performed using the RBF kernel, the parameters set are Cost (C) and Gamma ( $\gamma$ ). As in the linear kernel, parameter optimization is usually done through trial-and-error method. The following is the RBF kernel equation.

$$K(x_i, x) = \exp(-\gamma |X_i^T X|^2) \tag{2}$$

The RBF kernel It is employed to evaluate the similarity between two vectors  $x_i$  and  $x$  with the formula  $K(x_i, x) = \exp(-\gamma |X_i^T X|^2)$ . In this formula,  $|X_i^T X|^2$  represents the squared dot product of the two vectors, while the parameter  $\gamma$  controls the sensitivity of the kernel to the distance between vectors. The exponential function  $\exp$  ensures that the kernel value is ranged between 0 and 1, where a higher value is indicated to show greater similarity.

c. Polynomial Kernel

Cost (C) and Degree (d) are the optimal parameters in the polynomial kernel analysis. Similar to the linear and RBF kernels, these parameters are often optimized using a trial-and-error approach. The polynomial kernel equation is as follows.

$$K(x_i, x) = (\gamma \cdot X_i^T X + r)^p \tag{3}$$

The polynomial kernel is calculated using the formula, where  $X_i^T X$  is the dot product between the two vectors  $x_i$ , and  $x$ . The parameter  $\gamma$  controls the scaling factor, determining the influence of the dot product term, while  $r$  is a constant that allows the kernel to be adjusted as needed. The parameter  $p$  represents the polynomial degree, which controls the complexity of the decision boundary. The polynomial kernel is used to handle non-linear relationships in data, and parameters such as  $\gamma, r$ , and  $p$  are typically optimized through the distance between a class's vector points and the hyperplane is known as the margin.

d. Sigmoid Kernel

When the sigmoid kernel is used for analysis, the adjusted parameters include Cost (C) and Gamma ( $\gamma$ ). The selection of  $\gamma$  and C values needs to be done carefully so that errors do not occur in the results obtained. The following is the sigmoid kernel equation.

$$K(x_i, x) = \tanh(\gamma \cdot X_i^T X + r) \tag{4}$$

The sigmoid kernel is calculated using the formula  $K(x_i, x) = \tanh(\gamma \cdot X_i^T X + r)$ , where  $X_i^T X$  is the dot product between the two vectors  $x_i$  and  $x$ . The parameter  $\gamma$  controls the scale of the dot product, while  $r$  is a constant that adjusts the kernel function. The  $\tanh$  (hyperbolic tangent) function maps the result to a range between -1 and 1.

## 2.8 Random Forest

After Word2Vec feature extraction and data processing are finished, a random forest is used for classification, using several decision trees in the training stage. Random Forest is an extremely accurate and versatile technique. It is capable of handling tasks like regression and classification [20]. When it comes to training and making predictions, Random Forest operates very quickly. It also works well for managing big, noisy, high-dimensional datasets and gives information on how important each variable is to the classification process. Additionally, this approach can lower variability and is more resilient to issues that could come from boosting methods than a single decision tree [21]. Figure 6 shows an example of the Random Forest procedure.

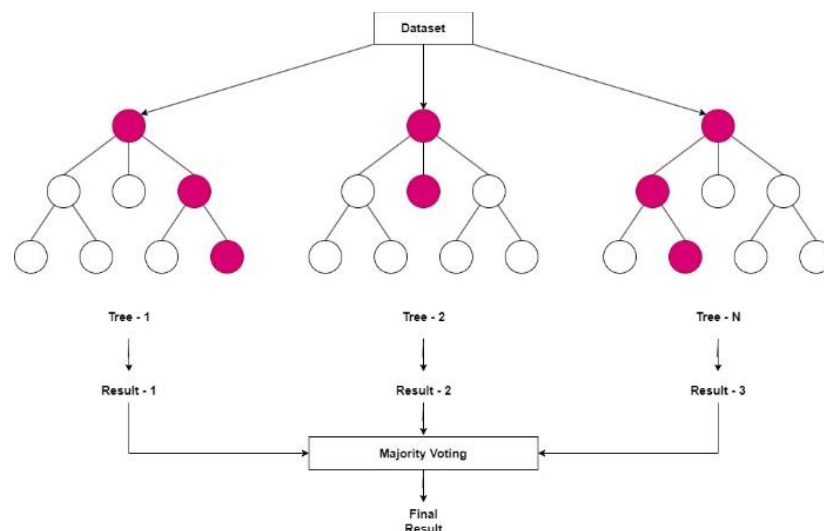


Figure 6. Random Forest Process

A few crucial hyperparameters must be adjusted for the Random Forest model to perform at its peak. Among these hyperparameters are `N_estimators`, which lower variance and increase model correctness and stability. `Max_depth` prevents overfitting while assisting in capturing the intricacy of the data. By taking into account distinct feature subsets for every split, `max_features` can lower correlation between trees. By ensuring that each split is only carried out if there are sufficient samples, the `min_samples_split` parameter can keep the tree from becoming overfitting and too specialized. Setting `min_samples_leaf` can enhance generalization and lessen overfitting, particularly in noisy datasets. How the split quality is measured can be determined by selecting the appropriate criterion.

## 2.9 Model Performance Measure

Precision, recall, accuracy, and F1-Score will be used in this study to evaluate the model's performance. The Confusion Matrix, the source of these four metrics, will be used to calculate the values for true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) [22]. The following formula can be used to formulate the Confusion Matrix processing:

### a. Precision

Precision is a metric for accuracy that indicates how effectively the system's information satisfies the needs of the user. Precision can be calculated using the formula below.

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

Accuracy It is computed as the ratio of True Positives (TP) to the sum of True Positives and False Positives (FP). The accuracy number shows the percentage of accurate findings among all results that the system deems significant.

### b. Recall

The system's ability to locate and return pertinent information is measured by a parameter called recall. Recall can be computed using the formula given.

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

The recall is calculated by dividing the number of True Positives (TP) by the sum of True Positives and False Negatives (FN).

### c. Accuracy

The degree to which the prediction resembles the actual value is known as accuracy. Better system predictions are indicated by a higher accuracy number. The following formula can be used to determine accuracy:

$$Accuracy = \frac{\text{correct prediction data}}{\text{Total data}} \quad (7)$$

$$= \frac{TP+TN}{TP+TN+FP+FN}$$

By dividing the total number of data points (True Positives + True Negatives + False Positives + False Negatives) by the number of accurate predictions (True Positives + True Negatives), accuracy is calculated.

### d. F1-Score

F1-score (F1) is calculated as the average of the precision and recall values.

$$F1 = 2 \times \frac{(Precision \times recall)}{(Precision + recall)} \quad (8)$$

The F1-score is the harmonic mean of precision and recall. To compute it, divide the result by the sum of the precision and recall, then multiply the precision and recall product by two.

Additionally, a cross-validation step will be used to increase the evaluation's reliability and lower the possibility of overfitting. In cross-validation, the dataset is divided into folds, a subset of the data is used to train the model, and another subset is used to assess the model's performance. By ensuring that model evaluation is independent of a single dataset split, this procedure produces a more reliable and consistent performance image.

## 3. RESULT AND DISCUSSION

There were several phases to this investigation. 9059 data points were preprocessed in the first step, which included cleaning, case folding, stopword removal, stemming, and tokenization. Split data is the following step, in which the data is separated into two sections: The Word2Vec Corpus uses 8059 data, and the model training and testing uses 1000 data. The data is divided with a ratio of 80% for training data and 20% for test data. Then, the training data and testing data will undergo the Word2Vec skip-gram feature extraction process with 100 dimensions and 300 dimensions. Next, the data will enter the classification model building stage using the SVM and RF methods which will end with an evaluation using the confusion matrix. There are several test scenarios performed, which are described in Table 8:

**Table 8.** Testing Scenario

Testing Scenario
<b>Scenario 1:</b> Seeing the effect of using the stemming process on the dataset
<b>Scenario 2:</b> Comparing the effect of using the number of dimensions of Word2Vec
<b>Scenario 3:</b> Comparing the results of all SVM kernels
<b>Scenario 4:</b> Comparing results on Random Forest with different parameters

In table 8, there are 4 scenarios that will be carried out in this study. The first example is to observe how the stemming process is used in preprocessing. The second scenario is to compare the use of Word2Vec with a dimension of 100 and a dimension of 300. The third scenario is to compare the results of all SVM kernels with the combination of the previous scenarios. And the fourth scenario is to compare the RF results with the combination of scenario 1 and scenario 2.

### 3.1 Scenario 1 Testing Results

Testing is done in scenario 1 to see the effect of using stemming and without stemming in the preprocessing stage. This testing uses 300 dimensions in Word2Vec and SVM with the RBF kernel as a sample. The output of this scenario is a dataset with stemming and a dataset without stemming. The results of scenario 1 can be seen in Table 9.

**Table 9.** Result of Testing the Effect of Stemming Process

Preprocessing	Accuracy	Precision	Recall	F1-score
<b>Stemming</b>	88.5%	88.7%	88.5%	88.5%
<b>No Stemming</b>	84.9%	85.5%	84.9%	85%

Based on the results of scenario 1, testing datasets using stemming have higher accuracy, precision, recall, and f1-score values compared to datasets that do not use stemming, where stemming has the highest accuracy value of 88.5%. This can happen because the quality of text representation in natural language processing can be improved through the stemming process, which is the mapping and decomposition of various forms (variants) of a word into its basic word form. Without stemming, it has the highest result of 84.9%. Where when compared to datasets that go through the stemming process is a lower value. This proves that the use of stemming can improve the final performance.

### 3.2 Scenario 2 Testing Results

Scenario 2 was tested to compare the number of dimensions used. This research uses 100 dimensions and 300 dimensions with the skip-gram algorithm on Word2Vec. The dataset used is a dataset that has passed scenario 1, which uses stemming, with SVM kernel RBF as a test sample. The results of scenario 2 can be seen in Table 10.

**Table 10.** Test Results for Dimension Usage in Word2Vec

Word2Vec	Accuracy	Precision	Recall	F1-score
<b>100 dimensions</b>	87.8%	88.1%	87.8%	88.5%
<b>300 dimensions</b>	88.5%	88.7%	88.5%	88.5%

Based on the results of scenario 2, the use of dimension 300 in Word2Vec has higher results compared to the use of dimension 100. The highest accuracy result at dimension 300 is 88.5% which is only 0.7% different from the accuracy result of dimension 100. This is because larger dimensions can capture semantic and syntactic relationships between words more richly. High dimensional vectors have a greater capacity to represent complex contextual and semantic information, thus reducing ambiguity in understanding word meaning [23]. This research can conclude that the use of 300 dimensions can improve the performance value of the Word2Vec skip-gram model.

### 3.3 Scenario 3 Testing Results

Testing scenario 3 is done to compare the results of all kernels in SVM whose datasets are obtained from the results of scenario 1 and scenario 2, namely using stemming and Word2Vec skip-gram with 300 dimensions. This test will compare the results of polynomial, RBF, sigmoid, and linear kernels. The results of scenario 3 can be seen in Table 11.

**Table 11.** Testing Results Using SVM

SVM	Accuracy	Precision	Recall	F1-Score
<b>Polynomial</b>	84%	85.8%	84%	82.8%
<b>RBF</b>	88.5%	88.7%	88.5%	88.5%



<b>Sigmoid</b>	75.2%	76.7%	75.2%	75.6%
<b>Linear</b>	83.6%	84.3%	83.6%	83.8%

Based on the results of scenario 3, a comparison of the performance of polynomial and linear kernels shows little difference although the polynomial kernel is slightly better. Polynomial kernels handle non-linear relationships in text data more effectively than linear kernels, which tend to perform better on data with simpler, linear patterns. The polynomial kernel, with its ability to capture more complex relationships, slightly improved accuracy performance over the linear kernel (84% vs. 83.6%). However, this difference is not very significant and shows that the linear kernel still performs quite well on this dataset. The SVM method with the RBF kernel demonstrates superior performance compared to the polynomial, sigmoid, and linear kernels. Where the RBF kernel has an accuracy value of 88.5%, precision 88.7%, recall 88.5%, and f1-score 88.5%. This is because RBF can handle non-linear data in text and its ability to capture complexity and variation in text data better than other kernels [24]. So, it can be concluded that the use of SVM with RBF kernel is more efficient than other SVM kernel models.

### 3.4 Scenario 4 Testing Results

Scenario 4 was tested to compare the results of several parameters in Random Forest. Where the dataset is obtained from the results of scenario 1 and scenario 2, namely using stemming and Word2Vec skip-gram with 300 dimensions. In this test, we will compare the results of Random Forest with several different parameters. The results of scenario 4 can be seen in Table 12.

**Table 12.** Testing Results Using RF

Random Forest	Parameter	Accuracy	Precision	Recall	F1-Score
<b>n_estimators=100 max_depth=20 max_features='sqrt' min_samples_split=5 min_samples_leaf=2 criterion='gini'</b>	1	85.8%	85.6%	85.8%	85.6%
<b>n_estimators=100 max_depth=20 max_features='sqrt' min_samples_split=5 min_samples_leaf=2 criterion='entropy'</b>	2	86%	85.9%	86%	85.8%
<b>n_estimators=200 max_depth=20 max_features='sqrt' min_samples_split=10 min_samples_leaf=4 criterion='gini'</b>	3	86.3%	86.1%	86.3%	86.1%

The results from scenario 4 show that the Random Forest model produced the highest accuracy (86.3%), precision (86.1%), recall (86.3%), and F1-score (86.1%). When parameters were set with n\_estimators set to 200 provides a balance between complexity and generalization, reducing the risk of overfitting. Max\_features set to 'sqrt', and min\_samples\_split set to 10 helps prevent the model from relying too heavily on specific features or data, preserving the model's ability to generalize without losing important information. This happens because the combination of parameters provides a balance between model complexity and generality. This allows the model to capture important patterns in the data without overfitting and results in the best performance in the evaluation metrics.

### 3.5 Discussion

By conducting several test scenarios, the test results of using stemming in preprocessing (scenario 1) improve the performance of the model. This is because stemming simplifies text representation, reduces noise, and improves semantic understanding. In addition, scenario 2, which compares the effect of using the number of dimensions of Word2Vec, is also used as a baseline to evaluate the performance of the text representation. At a dimension of 300, the skip-gram algorithm performed better than at a dimension of 100, with the highest accuracy of 88.5%, as the larger dimension was able to represent richer semantic relationships and reduce ambiguity. Meanwhile, scenario 3 shows that the influence of SVM with RBF kernel gives the highest results due to its ability to handle non-linear data and complex variations in text. While the optimal parameter combination in RF (n\_estimators=200, max\_depth=20, max\_features='sqrt', etc.) in scenario 4 gave the best accuracy of 86.3%, with a balance between complexity and generalization, thus avoiding overfitting. One of the main limitations of this study lies in the dataset used for this research. Although the preprocessing and splitting process aims to ensure fair representation of the data, potential biases such as unbalanced representation of positive and negative reviews can still occur. If one class is over-

represented, this can lead to skewed models that perform better for the majority class while underperforming for the minority class. In addition, variations in review length can pose challenges in text representation. Short reviews may not have enough contextual information for Word2Vec embedding to capture meaningful relationships, while excessively long reviews may introduce noise and make feature extraction less effective. The overall analysis shows the importance of selecting appropriate preprocessing, text representation, algorithms, and parameters to improve model performance.

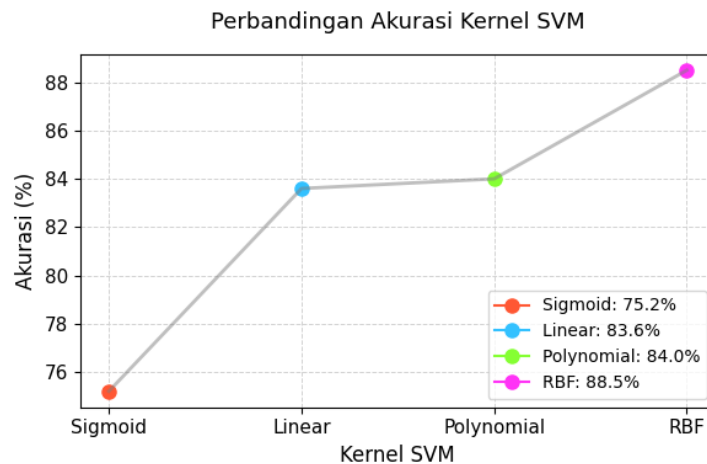


Figure 7. SVM Model Accuracy with Different Kernels

The SVM algorithm's kernel comparison in Figure 7 demonstrates that kernel choice significantly affects model accuracy. With an accuracy of 84%, the linear kernel outperforms the sigmoid kernel by 8%, which was previously just 76% accurate. With an accuracy of 84.5%, the polynomial kernel, meanwhile, offered a minor boost. The RBF kernel outperformed the polynomial kernel by 4%, achieving the best ultimate accuracy of 88%. These outcomes demonstrate how well the RBF kernel handles data with non-linear trends.

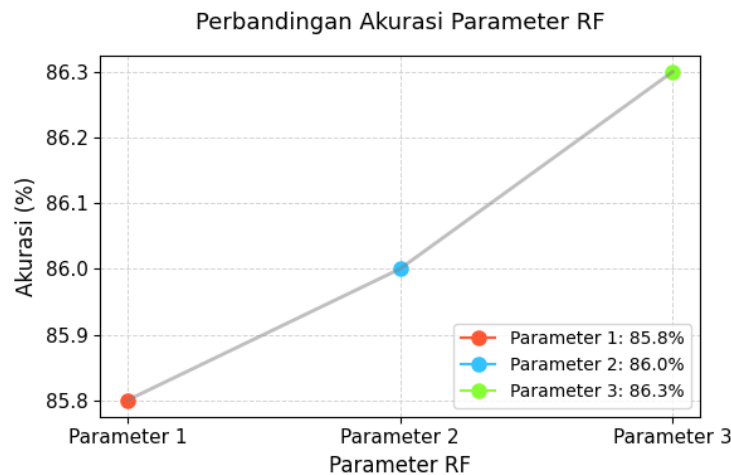


Figure 8. Accuracy Comparison Based on RF Parameters

Figure 8 shows how the Random Forest (RF) algorithm's parameter changes significantly affect the accuracy of the model. With a 0.5% increase, Parameter 3 produced the maximum accuracy of 86.3%, while Parameter 1 produced an accuracy of 85.8%. This improvement implies that selecting optimal parameters, such as the number of trees ( $n\_estimators$ ), the maximum depth of the tree ( $max\_depth$ ), and the number of randomly selected features ( $max\_features$ ), can enhance the model's ability to recognize complex patterns in the data while maintaining generalization for better performance.

Even while SVM with RBF kernel outperformed Random Forest in terms of accuracy (88.5%), a number of factors influence which method is better. Although Random Forest can handle larger datasets more quickly and is parallelizable, it is more efficient in terms of training time than SVM in terms of accuracy. In contrast, SVM performs better on non-linear and high-dimensional datasets. In terms of dataset compatibility, Random Forest's ensemble nature, which guards against overfitting, makes it more resilient to noisy and unstructured data. Because of this, the decision between SVM and Random Forest relies on certain factors including dataset size, computing time, and data complexity.

## 4. CONCLUSION

Based on this research, a sentiment analysis program for Photomath app user reviews has been built using Random Forest and SVM methods with Word2Vec feature extraction. The dataset used consisted of 9059 review data, which was evaluated through various preprocessing methods, including stemming, as well as comparison between Word2Vec and 300 dimensions using skip-gram algorithm. The results showed that SVM with RBF kernel consistently produced the best performance with 88.5% accuracy. The RBF kernel's ability to handle high-dimensional word vectors makes it more resilient to variations in preprocessing. These findings suggest that the RBF kernel is effective for sentiment classification tasks, especially when using Word2Vec with large dimensions. The practical implications of this research can improve Photomath's functionality by helping developers to better understand and address common user concerns, which in turn increases application user satisfaction. Although SVM with RBF kernel achieved the highest accuracy, Random Forest remains a valuable method, especially for tasks that require simpler interpretability. Some of the challenges faced in this research, such as the computational complexity and size of the Word2Vec model, can be overcome by exploring more lightweight feature extraction techniques. Future research could also expand the dataset with other languages to create a more universal sentiment analysis system. Thus, increasing the dimensionality of Word2Vec or using alternative vectorization methods can further improve accuracy, while further analysis of the influence of stopwords is also important to reduce their influence on model performance. Future research should focus on system optimization to reduce execution time and improve scalability of real-time analysis.

## REFERENCES

- [1] M. F. L. Sibuea, M. A. Sembiring, I. Almeina, and R. T. A. Agus, "Pemanfaatan Aplikasi Photomath Sebagai Media Belajar Matematika," *J. Pemberdaya. Sos. dan Teknol. Masy.*, vol. 2, no. 1, p. 109, 2022, doi: 10.54314/jpstm.v2i1.962.
- [2] I. N. M. Zain, M. A. B. Setambah, M. S. Othman, and M. H. M. Hanapi, "Use of Photomath Applications in Helping Improving Students' Mathematical (Algebra) Achievement," *Eur. J. Educ. Pedagog.*, vol. 4, no. 2, pp. 85–87, 2023, doi: 10.24018/ejedu.2023.4.2.601.
- [3] A. Car *et al.*, "Analisis Sentimen Review Pengguna Aplikasi Photomath Dengan Metode Support Vector Machine (SVM)," *Int. J. Technol.*, vol. 47, no. 1, p. 100950, 2023.
- [4] V. D. I. S. Cilegon, "Penerapan pemanfaatan multimedia pada aplikasi photomath dalam pembelajaran trigonometri kelas xi mipa v di sman 3 cilegon," pp. 289–296, 2023, doi: 10.57254/ijtl.v1i3.45.
- [5] F. Zahra and Y. Yahfizham, "Systematic Literature Review : Memanfaatkan Aplikasi Photomath Sebagai Media Belajar Untuk Meningkatkan Kemampuan Komputasi Siswa," *Bilangan J. Ilm. Mat. Kebumian dan Angkasa*, vol. 2, no. 3, pp. 26–32, 2024, [Online]. Available: <https://doi.org/10.62383/bilangan.v2i3.46>
- [6] F. Syafira, "Analisis Sentimen Dampak Perkembangan Artificial Intelligence (AI) Pada Media Sosial Twitter Menggunakan Metode Support Vector Machine Dan Lexicon Based," *Institutional Repos. UIN Syarif Hidayatullah Jakarta*, p. 163, 2023.
- [7] I. F. Rozi, A. T. Firdausi, and K. Islamiyah, "Analisis Sentimen Pada Twitter Mengenai Pasca Bencana Menggunakan Metode Naive Bayes Dengan Fitur N-Gram," *JIP (Jurnal Inform. Polinema)*, vol. 6, pp. 33–39, 2020, doi: <https://doi.org/10.33795/jip.v6i2.316>.
- [8] A. A. Firdaus *et al.*, "Application of Sentiment Analysis as an Innovative Approach to Policy Making: A Review," *J. Robot. Control*, vol. 5, no. 6, pp. 1784–1798, 2024, doi: 10.18196/jrc.v5i6.22573.
- [9] M. J. Aufa and A. Qoiriah, "Analisis Sentimen Pengguna Platform Belajar Online Coursera menggunakan Random Forest dengan Metode Ekstraksi Fitur Word2vec," *(Journal Informatics Comput. Sci. ISSN)*, vol. 04, pp. 244–255, 2022, doi: <https://doi.org/10.26740/jinacs.v4n02.p244-255>.
- [10] F. A. Larasati, D. E. Ratnawati, and B. T. Hanggara, "Analisis Sentimen Ulasan Aplikasi Dana dengan Metode Random Forest," *J. Pengemb. Teknol. Inf. dan Ilmu Komput.*, vol. 6, no. 9, pp. 4305–4313, 2022.
- [11] S. N. Adhan *et al.*, "Analisis sentimen ulasan aplikasi wattpad di google play store dengan metode random forest," *Anoatik J. Teknol. Inf. dan Komput. Vol. 2, No. 1, Juni 2024 e-ISSN*, vol. 2, no. 1, pp. 6–15, 2024, doi: <https://doi.org/10.33772/anoatik.v2i1.32>.
- [12] R. Supriyadi, W. Gata, N. Maulidah, and A. Fauzi, "Penerapan Algoritma Random Forest Untuk Menentukan Kualitas Anggur Merah," *E-Bisnis J. Ilm. Ekon. dan Bisnis*, vol. 13, no. 2, pp. 67–75, 2020, doi: 10.51903/e-bisnis.v13i2.247.
- [13] M. A. A. Jihad, Adiwijaya, and W. Astuti, "Analisis Sentimen Terhadap Ulasan Film Menggunakan Word2Vec dan SVM," *e-Proceeding Eng.*, vol. 8, no. 4, pp. 4136–4144, 2021.
- [14] D. I. Rifai, "Implementasi Word2Vec Pada Analisis Sentimen Terhadap Ulasan Pengguna Aplikasi Tiktok Menggunakan Metode Support Vector Machine," 2024, [Online]. Available: <http://repository.unissula.ac.id/id/eprint/34039>
- [15] P. E. Shoppee and S. Watmah, "Komparasi Metode K-NN , Support Vector Machine , Dan Random Forest," *Insa. – J. Inov. dan Sains Tek. Elektro*, vol. 2, no. 1, pp. 15–21, 2021, doi: 10.31294/instk.v2i1.419.
- [16] E. Engineering, "Analisis Sentimen Terhadap Penggunaan Aplikasi Shopee Menggunakan Algoritma Support Vector Machine (SVM)," *Jambura J. Electr. Electron. Eng.*, vol. 5, pp. 32–35, 2023, doi: <https://doi.org/10.37905/jjee.v5i1.16830>.
- [17] B. W. Sari and F. F. Haranto, "Implementasi Support Vector Machine Untuk Analisis Sentimen Pengguna Twitter Terhadap Pelayanan Telkom Dan Biznet," *J. Pilar Nusa Mandiri*, vol. 15, no. 2, pp. 171–176, 2019, doi: 10.33480/pilar.v15i2.699.
- [18] D. Intan, S. F. Handayani, and R. W. Pratiwi, "Pengaruh Parameter Word2Vec terhadap Performa Deep Learning pada Klasifikasi Sentimen," vol. 6, no. 3, pp. 156–161, 2021, doi: <https://doi.org/10.30591/jpit.v6i3.3016>.
- [19] P. Fremmuzar and A. Baita, "Uji Kernel SVM dalam Analisis Sentimen Terhadap Layanan Telkomsel di Media Sosial Twitter," *Komputika J. Sist. Komput.*, vol. 12, no. 2, pp. 57–66, 2023, doi: 10.34010/komputika.v12i2.9460.
- [20] N. G. Ramadhan and F. D. Adhinata, "Teknik Smote Dan Gini Score Dalam Klasifikasi Kanker Payudara," *RADIAL J. Perad. Sains, Rekayasa dan Teknol.*, vol. 9, no. 2, pp. 125–134, 2021, doi: 10.37971/radial.v9i2.229.



- [21] F. I. Rafif, M. D. Purbolaksono, and ..., “Sentiment Analysis using Random Forest and Word2Vec for Indonesian Language Movie Reviews,” *J. Media ...*, vol. 7, pp. 1109–1116, 2023, doi: 10.30865/mib.v7i3.6299.
- [22] E. Suryati, Styawati, and A. A. Aldino, “Analisis Sentimen Transportasi Online Menggunakan Ekstraksi Fitur Model Word2vec Text Embedding Dan Algoritma Support Vector Machine (SVM),” *J. Teknol. Dan Sist. Inf.*, vol. 4, no. 1, pp. 96–106, 2023, [Online]. Available: <https://doi.org/10.33365/jtsi.v4i1.2445>
- [23] K. Rahmata, “Implementasi metode word2vec dan vector space model pada sistem temu kembali informasi pembelajaran sirah nabawiyah,” *Repository.Uinjkt.Ac.Id*, 2020, [Online]. Available: <https://repository.uinjkt.ac.id/dspace/handle/123456789/55997>
- [24] M. M. Siregar, R. Hizria, and D. Pardede, “Perbandingan Kinerja Kernel SVM dalam Klasifikasi Kategori Kanker Kulit Menggunakan Transfer Learning,” vol. 4, no. 1, pp. 83–90, 2024, doi: <https://doi.org/10.47709/dsi.v4i1.4665>.